Alexey Cherchago
Dissertation
"Service Specification and Matching based on Graph Transformation"

Abstract

One of the main tasks of modern application integration projects is to allow one business unit (requestor) to use services offered by another business unit (provider). When software systems of business partners are composed, an import interface of the requestor system is connected to an export interface of the provider system. Here, the import interface specification containing the requestor's requirements for a needed service has to be matched against the export interface specification describing the provided service. Usually, software engineers carry out matching of interface specifications manually; it makes the design and implementation of composite software expensive and error-prone. Therefore, the demand for instruments that would automate the matching procedure is high.

In this thesis, we develop a new technique facilitating integration of software systems. To this end, we solve a problem of how to construct visual and formal interface specifications comprising *semantic* descriptions. Our method also includes a matching procedure that checks compatibility of such specifications.

Interface specifications consist of structural and behavioural compartments. The structural compartment, given by a signature analogous to those appearing in algebraic specifications, defines operation declarations. The behavioural compartment, modelled by a conditional graph transformation system (GTS), contains operation contracts in the form of graph transformation rules. The rules of conditional GTS are equipped with loose semantics to describe operations in the import interface, and with strict semantics to describe those in the export interface. Composition of two compartments leads to an integral interface specification which is represented by the novel concept of *parameterized conditional GTS*.

We develop three kinds of compatibility relations underlying the matching procedure. The intended correspondence between declarations and contracts of the required and provided operations is reflected by *structural* and *behavioural* compatibility relations that are established over the corresponding compartments of interface specifications. These two compatibility relations are combined into an *integral* compatibility relation that links the integral specification of the import interface to the one of the export interface. Furthermore, the constructed relations are equipped with rigorously formulated semantic requirements to compatibility and are justified against them.

The introduced mathematical theory is supplemented with a *conceptual framework*. It is aimed at generating interface specifications that are suitable for automation of the matching process. The framework is based on an industry standard that outlines a uniform way of generating specifications. We use the standard issued by the Open Travel Alliance (OTA) in our example scenario where we develop and match standard-based interface specifications of Web services taken from the travelling business domain.

Compatibility of interface specifications is necessary but not sufficient for accurate interactions between systems. The integration process is based on the assumption that these systems are correct. First and foremost, this correctness means that interface specifications representing externally visible parts of systems are consistent with implementations which appear internally in the systems. To check this assumption, we propose a model describing external as well as internal parts of a system. The model, formally represented by a *graph transformation module*, defines consistency relations between external and internal specifications and allows to validate correctness of systems prior to the integration. The proposed model and the matching procedure developed in the thesis are the key elements of a technology designed to improve the application integration process, making it theoretically well-defined and practically machine-processable.